Introducing Product Line Architectures in the ERP Industry:
Challenges and Lessons Learned

Haitham S. Hamza
Software Engineering Competence Center
Cairo University
Giza 12613, Egypt
hshamza@acm.org

Jabier Martinez and Carmen Alonso
European Software Institute (ESI-Tecnalia)
Parque Tecnológico de Bizkaia #204
E-48170 Zamudio, Spain
{ jabier.martinez | carmen.alonso } @esi.es

Abstract—Return on Investment (ROI) for companies involved in Enterprise Resource Planning (ERP) system development depends on their flexibility to evolve, maintain, customize and configure their ERP product to respond to new business needs, deployment models and emerging market segments. In this particular aspect, ERP systems can get benefit from commonality and variability management concepts in order to improve evolution and maintainability. Moreover, Product Line Engineering (PLE) methods and practices can substantially reduce time and effort regarding the current complex and tedious configuration procedures that are not only resource-intensive, but also error-prone. This paper introduces practical experiences from the application of product line architectures (PLAs) in four companies of the ERP systems domain.

Keywords-Software Product Lines; Enterprise Resource Planning systems; Systematic reuse

I. INTRODUCTION

The main goal for ERP systems adoption is to boost the performance of the organizations’ resource planning, management control and operational control by integrating activities across functional departments including planning, manufacturing, purchasing, controlling and maintaining inventory, tracking orders and so on. Return on Investment (ROI) for companies that develops ERP systems depends on their ability to evolve, maintain, and customize or configure their own ERP to respond to new business needs, deployment models and emerging market segments.

ERP system development fits naturally into the thinking philosophy of PLE as these systems can benefit greatly from the concepts of commonalities and variability management to enhance evolution and maintainability of their modules.

The objective of this paper is to communicate to the community practical experiences from applying the concept of Product Line Architectures (PLA) into real-life projects in the ERP systems domain.

This paper is structured as follows. First, an overview of the scope of the case-studies and the implemented methodology is provided. Second, the main challenges and the learned lessons are discussed separated in different categories. Finally, some conclusions about the findings are presented.

II. PROJECT OVERVIEW

The results presented in this paper are obtained during a Cluster Approach Project with four ERP major companies in Egypt. This project was coordinated by SECC [1] in collaboration with ESI-Tecnalia [2] with the objective to improve the quality of the selected companies’ software development focusing on the improvement or application of systematic reuse practices. ESI-Tecnalia has been working for more than ten years in different systematic reuse technologies and it offers products and services covering all the phases of systematic reuse transition, from awareness to full implementation. ESICenter SECC established by SECC in cooperation with ESI-Tecnalia has experts authorized by ESI-Tecnalia to provide these services.

Each of the mentioned projects was conducted by a task force of three SECC consultants, four engineers and consultants from ESI-Tecnalia and a company specific team composed by business managers, technical managers and software developers. At the end of these projects the participating SMEs had a trained team in setting up a systematic reuse strategy addressing the specific business needs, with practical experience in executing a pilot within the company and having assessed the benefits obtained from SPL technology adoption. All projects were structured in a similar fashion, with three initial phases executed with all SMEs in a joint fashion and a final phase executed per-SME.

The following list enumerates the executed phases of the Reuse-Cluster Approach Project.

- Phase 1 – Awareness: A seminar that explains the project structure and the expected benefits (open to all SMEs interested in joining the project now or in the future)
- Phase 2 – Diagnosis and Planning: A workshop to diagnose current situation and then produce a specific strategic plan for each of the SMEs.
- Phase 3 – Training: Hands-on training for the technical team that will deploy the reuse strategy in each SME, exploring ESI-Tecnalia SPL technologies.
- Phase 4 – Implementation and Pilot: Implementation of the first steps of the reuse strategy in each SME and pilot in a real project to assess initial results.
The Product Line Unified Modeller (PLUM) [3] was the tool suite used to model a specified scope in each ERP product to show the benefit of PLAs in developing customized ERPs.

The four companies tried to solve the challenges of PLE using their own approaches. Nevertheless, they experiment some degree of discomfort with their current situation and there existed management commitment to face a change. Some examples of the discomfort regarding their previous situation are outlined as follows:

- Ad-hoc reuse practices leading to over-engineered solutions that meet the market needs but require constant investment to ensure evolution and maintainability.
- Challenging relationship between marketing and engineering teams due to engineering’s inability to accommodate business requests for more product variations.
- Complex and time consuming coordination of multiple parallel development efforts that may share common objectives.
- Lack of ERP product knowledge encapsulation that made difficult knowledge transition.

### III. CHALLENGES IN THE ERP SYSTEMS DOMAIN

The following subsections summarize some of the key challenges observed during the project. Most of these challenges are relevant in other domains. The first one is probably the most specific for the ERP domain.

**A. No differentiation between customization and configuration**

ERPs use to have many features for allowing the user to configure the system. From a PLE point of view, variability is not something that varies in the same way for all the products. It was observed that the companies have problems to differentiate between product variability and this kind of customization. Given that they do not clearly differentiate between product configuration and customization, they use the same development pattern: ERP product has the ability to interpret at runtime some type of configuration metadata. They have more or less sophisticated methods for ERP configuration, but none for ERP product derivation so they use to provide to their customers most of the compiled ERP code. As long as the variability increases, the system becomes more difficult for the developers to manage, that has a serious influence on the scalability of the final ERP. This makes the starting point for the ERP Company to realize the added value of product derivation process and the importance in differentiating between “Customization rules” and “System rules”. As consequence they started identifying variation points inside components.

**B. Coarse granularity of variability**

Because of the pressure to respond answer quickly to a concrete customer needs and due to a lack of a clear modularization strategy, the ERP companies tend to group design decisions that actually contains many ERP design decisions. This leads towards the following key problems:

- **Hidden Knowledge**: the knowledge about features is not easy to share with others. As a result, individuals who participated in the requirements analysis and/or development of a particular feature are the only ones who know about the functionalities of this particular feature.
- **Redundancy**: Due to the difficulty to share this hidden knowledge, it is very likely that a new requirement raised by a customer has been already implemented somewhere in one of the features, but the feature may be implemented again from scratch, wasting time and resources.
- **Inefficiency**: the binary decision of activating or deactivating this kind of features leads to coarse reuse granularity, as features usually contain much functionality that are not needed by all customers. Consequently, the generated product ends up having extra unused (and indeed not needed) functionalities. Moreover, by deactivating a feature, the ERP may not be able to reuse a particular functionality within the feature, and thus, this particular functionality (or set of functionalities for this matter) may need to be implemented from scratch or reused in an ad-hoc fashion.

**C. System knowledge based on individuals**

ERP systems are complex systems whose understanding should not rely on companies’ individuals. This is risky as knowledge could be lost and it is also expensive when this information has to be transferred. Lack of useful documentation and inaccuracies were found during the projects. PLA and variability management systems as PLUM allow ERP systems knowledge encapsulation.

**D. Unclear mapping between requirements and features**

Reuse through product line enhanced requirements analysis activities. It was observed that most of companies lack a systematic way to efficiently gather the requirements.
of their customers. This is because, the mapping between the requirements and the functionalities is not clear. Using PLA (i.e. using the PLUM tool), more systematic requirement analysis has been possible.

E. Development environment not suitable for automation

A limitation for PLA adoption in ERP systems domain, as well as in other industry settings, consists on development environment’s lack of automation capabilities. Adopting Systematic Reuse within companies’ frameworks and used technologies could represent a very expensive integration effort that has to be taken into account in early stages.

F. Complex re-engineering process

A major limitation for existing ERP systems that face the adoption of a PLA is the re-engineering process. Cost, time, and effort are typically the main barriers. Starting a bottom-up product line development approach demanded a high investment for making the assets reusable in a proper way.

However, this re-engineering process could be incrementally done while creating the PLA. During one of the pilot projects that involved legacy code, reusable assets for the derivation of four ERP modules were implemented with a reduction of the 54% of the originally needed Lines of Code with the PLA adoption with PLUM.

IV. LESSONS LEARNED

The challenges and issues observed throughout the four pilot projects are more or less typical in PLA projects in most domains. However, some new ones have been detected thanks to the pilot projects regarding the particular domain of the ERP systems.

A. Adapt the PLA to their perspective of an ERP system

Although the projects were in the same domain, companies’ perspectives were different in how they perceive their own ERP products. It was observed two particular patterns for ERP systems perspectives: process-oriented and business-oriented. Both of them are complementary views: ERP business-oriented modules support ERP processes and at the same time, ERP process-oriented are mapped to ERP business modules. However, this aspect conditioned their expectations for the PLA and in this sense PLA adoption cannot impose an architectural perspective for their products. The proposed PLA should be adapted to their ERP systems point of view.

B. ERP system configuration decision-flow pattern

The decision-flow for an ERP system configuration is defined as the variability resolution process used to build a concrete ERP configuration for a customer. The decision-flow implemented in PLUM for the four companies consists of a set of interrelated decisions that guide the Application Engineer, eventually in collaboration with the customer, through a suitable ERP configuration. Special attention was paid to the design and definition of a pattern for a decision-flow based on the thinking process of the companies’ domain experts.

<table>
<thead>
<tr>
<th>Decision-Flow depending on ERP company type</th>
<th>Business-oriented</th>
<th>Process-oriented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain</td>
<td>Domain processes</td>
<td></td>
</tr>
<tr>
<td>Modules</td>
<td>Processes</td>
<td></td>
</tr>
<tr>
<td>Sub-Modules</td>
<td>Steps</td>
<td></td>
</tr>
<tr>
<td>Features</td>
<td>Details</td>
<td></td>
</tr>
</tbody>
</table>

The four companies agreed that the decision-flow pattern for ERP configuration needs to take into account the customer perspective first, since this decision directly affects what is more suitable for the customer, based on their ERP configurations experience.

For simplifying the variability resolution process, dependencies among decisions should be triggered affecting only decisions of the same level or lower levels. For example, the decision to use certain feature of a Sub-Module should not affect the selection of another Module. This decision-flow can greatly affect the complexity as well as the efficiency of developing a PLA for an ERP system.

V. CONCLUSIONS AND FURTHER WORK

ERP systems development companies can take great competitive advantages adopting Systematic Reuse approaches supported by state-of-the-art PLE methods, practices and tools. The four companies mentioned in this paper saw the Systematic Reuse adoption potential to open new business opportunities, encapsulate company’s product knowledge and improve their development processes. They showed interest to continue with the approach in order to solve the presented challenges, and currently, extensions to the pilot projects are underway.

From a further research point of view, commonality among the four companies reveals that, there is a higher level of commonality in ERP systems that can be exploited and modeled at a higher abstraction level. This abstract reuse process could be generalized to the ERP domain and instantiated and customized to meet business strategies and vision.

REFERENCES